

Claims

1. A computer-implemented method of pseudo-randomly selecting game elements for use in playing a game, wherein the game is played by a set of users, wherein the game elements are used to determine an outcome of the game, and wherein the method is performed by an oracle and comprises:

obtaining a set of seed data items, wherein the set of seed data items comprises one or more user seed data items generated by a respective user;

obtaining a sequence of public keys;

obtaining a list of game elements, wherein a total number of public keys corresponds to a total number of game elements; and

generating a first output of a game transaction, wherein the game transaction is a blockchain transaction, wherein the first output comprises the sequence of public keys, and wherein the output comprises a script configured to generate at least one pseudorandom number, the at least one pseudorandom number being based on the set of seed data items, wherein the script is configured to generate a list of the public keys based on the at least one pseudorandom number, and wherein an order of public keys in the list of public keys differs compared to an order of public keys in the sequence of public keys.

2. The method of claim 1, comprising transmitting the game transaction to one or more of the respective users and/or the blockchain network.

3. The method of claim 1 or claim 2, wherein the output script is configured to generate a plurality of pseudorandom numbers, and to generate the list of public keys based on one, some or all of the plurality of pseudorandom numbers.

4. The method of any preceding claim, wherein the output script is configured to generate the list of public keys by selecting, for each respective pseudorandom number, a public key at a position in the sequence of public keys corresponding to the respective pseudorandom number, and to place the selected public key at a beginning of the list of public keys.

5. The method of any preceding claim, comprising, generating a map, wherein the map comprises a mapping of public keys in the list of public keys to game elements in the list of game elements.
6. The method of any preceding claim, wherein the sequence of public keys comprises one or more first sets of public keys, and wherein at least one of the first sets of public keys is generated by a respective user.
7. The method of claim 6, wherein said obtaining of the sequence of public keys comprises obtaining the one or more first sets of public keys from the respective users.
8. The method of claim 6 or claim 7, wherein the sequence of public keys comprises a second set of public keys, and wherein the second set of public keys are generated by the oracle.
9. The method of any preceding claim, comprising:
generating, for each game element in the list of game elements, a respective proof token, wherein the proof token represents a respective position of the game element in the list of game elements.
10. The method of any preceding claim, comprising, generating a hash of the list of game elements.
11. The method of claim 10, wherein generating the hash of the list of game elements comprises:
generating a merkle tree, the merkle tree comprising a plurality of leaf node pairs, each leaf node pair comprising a first leaf node and a second leaf node, wherein each first leaf node in each leaf node pair is generated by applying a hash function to a respective game element, and wherein each second leaf node of each leaf node pair is generated by applying a hash function to a respective proof token, wherein the first leaf nodes are ordered according to the list of game elements.

12. The method of claim 11, comprising, generating a commitment transaction, wherein the commitment transaction is a blockchain transaction and comprises a root node of the merkle tree.
13. The method of claim 12, comprising, transmitting the commitment transaction to one or more of the respective users and/or the blockchain network.
14. The method of any preceding claim, wherein the set of seed data items comprises an oracle seed data item generated by the oracle.
15. The method of any preceding claim, wherein each respective pseudorandom number is generated by:
 - applying a respective hash function to a combination of the set of seed data items to generate a respective hash result; and
 - mapping the respective hash result to a number based on a total number of game elements in the list of game elements.
16. The method of claim 15, wherein said mapping of the respective hash result comprises taking a modulus of the respective hash result, wherein said total number is the modulus.
17. The method of claim 15 or claim 16, wherein applying each respective hash function comprises applying a same hash function a different number of times.
18. The method of claim 12 and claim dependent thereon, wherein the commitment transaction comprises the set of seed data items.
19. The method of claim 18, wherein the commitment transaction comprises a set of inputs, each respective input comprising a hash of a respective one of the set of seed data items.
20. The method of claim 9 or any claim dependent thereon, comprising:

to each of a first set of the respective users, transmitting a respective first set of the proof tokens and an indication of the respective game element at the respective position represented by each of the respective proof tokens.

21. The method of claim 9 or any claim dependent thereon, comprising:

to one or more of the first set of the respective users, transmitting one or more second sets of the proof tokens, and an indication of the respective game element at the respective position in the list of game elements represented by each of the respective proof tokens; and

to each of the one or more of the first set of the respective users and for each proof token in the one or more second sets of proof tokens, transmitting one or more sets of private keys, wherein each private key corresponds to a respective public key mapped to the respective game element at the respective position in the list of game elements represented by the respective proof token.

22. The method of claim 11, and claim 20 or claim 21, comprising:

to each respective user, transmitting a merkle path for each game element paired with each proof token transmitted to that respective user.

23. The method of any of claims 20 to 22, comprising:

generating a payout transaction, wherein the payout transaction is a blockchain transaction locked to a winning public key, and wherein the winning public key is generated based on a combination of public keys, each one of the combination of public keys mapped to a respective game element represented by a respective proof token transmitted to at least one of the first set of users.

24. The method of any preceding game, wherein the game is poker and wherein the game elements represent playing cards.

25. A blockchain transaction for inclusion in a blockchain, the transaction comprising:

an output, wherein the first output comprises a sequence of public keys, and wherein the output comprises a script configured to generate at least one pseudorandom

number, the at least one first pseudorandom number being based on a set of seed data items, wherein the set of seed data items comprises one or more user seed data items generated by a respective user, and wherein the script is configured to generate a list of public keys based on the at least one pseudorandom number to, wherein an order of public keys in the list of public keys differs compared to an order of public keys in the sequence of public keys, and wherein a total number of public keys corresponds to a total number of game elements.

26. A computer-readable storage medium having stored thereon the transaction of claim 25.

27. Computer equipment comprising:
memory comprising one or more memory units; and
processing apparatus comprising one or more processing units, wherein the memory stores code arranged to run on the processing apparatus, the code being configured so as when on the processing apparatus to perform the method of any of claims 1 to 24.

28. A computer program embodied on computer-readable storage and configured so as, when run on computer equipment of claim 26, to perform the method of any of claims 1 to 24.